



Auto OEMs: Don't Just Own the Software. Take Control of It.

Manage the Entire Software Lifecycle to Cut Costs, Improve User Experience, and Stay Competitive Despite Accelerating Change

WHEN IT MATTERS, IT RUNS ON WIND RIVER

EXECUTIVE SUMMARY

Automotive original equipment manufacturers (OEMs) are coming to grips with the fact that software is increasingly becoming the primary source of a vehicle's value today. Software now largely defines the consumer's experience. It determines how safely and reliably the car performs. It drives the instruments and infotainment systems the consumer interacts with, and much more. Software is the common control point for the three top trends impacting the auto industry—connectivity, electrification, and the rise of autonomous vehicles.

Given this, there is a growing urgency for OEMs to "own the software." However, owning the software is only a first step in keeping pace with the radical changes impacting the OEM business model today.

Yes, OEMs need to own the software; but, more importantly, they need to take total control of the software. They need to manage the entire software lifecycle—from ideation to deployment to updates—throughout the vehicle's life. By doing so, they'll open up many new opportunities for cutting costs, improving the user experience, and staying competitive in the face of accelerating change. This article explores the meaning, importance, and opportunity of software lifecycle management for auto OEMs today.

TABLE OF CONTENTS

Executive Summary	2
Owning vs. Managing Software: What's the Difference?	3
OTA: Key Element of Effective Software Lifecycle Management	3
The ACUR Model: A Software Strategy for Continuous Evolution	4
Solving Core Industry Challenges: Examples	5
Conclusion	5

OWNING VS. MANAGING SOFTWARE: WHAT'S THE DIFFERENCE?

Today a vehicle is no longer merely a physical object or a set of hardware specifications—it is a fast-evolving continuum of capabilities, delivered primarily by software. There are dozens of software-driven capabilities in today's vehicles, ranging from driver assistance features to in-vehicle infotainment (IVI) systems to safety and security features.

To "own" the software doesn't just mean owning the rights to the code (the intellectual property) that delivers the capabilities. Nor does it require creating a huge software team or buying a nimble software startup that can develop the next brilliant innovation. It means controlling the delivery of software capabilities as they evolve, today and tomorrow, across the lifecycle of the software.

"It is not the strongest of the species that survives, nor the most intelligent. It is the one that is most adaptable to change."

—Charles Darwin

In short, software is a living thing. From the perspective of the consumer, software-driven capabilities are constantly evolving to improve the driving experience, without any action required on the consumer's part. All the consumer sees is that things keep working better—more reliably, more safely, faster, with more choices, and so on.

From the OEM's perspective, software is dynamic because because it's a constant source of new opportunity to add value, reduce costs, meet evolving consumer demands, and enhance the brand image. For example, software could address new security threats without requiring a recall, add new infotainment or instrumentation capabilities without requiring a visit to the dealer, or provide exciting new features such as access to various cloud services from the road.

Equally important, software creates multiple opportunities to generate value as the vehicle ages and changes hands. Today the average vehicle is developed for 200,000+ equivalent customer miles,

and the average consumer owns the car 9–10 years and drives 10,000–15,000 miles per year. That means there are primary, secondary, and tertiary opportunities for an OEM to create customer satisfaction and loyalty by constantly improving the experience.

To put it simply, owning the software is necessary but it doesn't end there. Living things require management over time. OEMs have much to gain from mastering the art of software lifecycle management.

OTA: KEY ELEMENT OF EFFECTIVE SOFTWARE LIFECYCLE MANAGEMENT

It would be easy to assume that software lifecycle management is primarily about using over-the-air (OTA) software update capabilities for a broader range of use cases. After all, use of OTA is accelerating among forward-looking OEMs.

As cost pressures continue to mount from proliferating engine control units (ECUs) within the vehicle, OEMs are beginning to recognize the benefits of OTA technology for ECU consolidation, faster development and continuous software deployment cycles, postproduction enhancements, and reduced costs associated with feature-related recalls or fixes. Many auto OEMs are now targeting an increasing percentage of ECUs for OTA updates, including telematics, infotainment, and powertrain safety.

However, OTA is only one element of successful software lifecycle management.

From a broader perspective, software lifecycle management means controlling the evolution of the software everywhere it impacts the consumer experience. That begins with ideation—the moment OEMs start dreaming up new capabilities and new experiences they want to put into the vehicle—and it includes modeling those innovations, moving them into prototypes, creating an early production build, reusing and updating elements of the software to keep pace with evolving requirements or new capabilities, and monitoring and maintaining that vehicle and its software all the way through end of life.

All of that requires much more than a focus on OTA. It requires addressing four core capabilities: to abstract, consolidate, reuse, and update—or "ACRU."

THE ACRU MODEL: A SOFTWARE STRATEGY FOR CONTINUOUS EVOLUTION

The ACRU model recognizes that consumer demands for better, safer experiences through software will continue to evolve at an accelerating rate, and it creates a software strategy that fully addresses the need for continuous evolution.

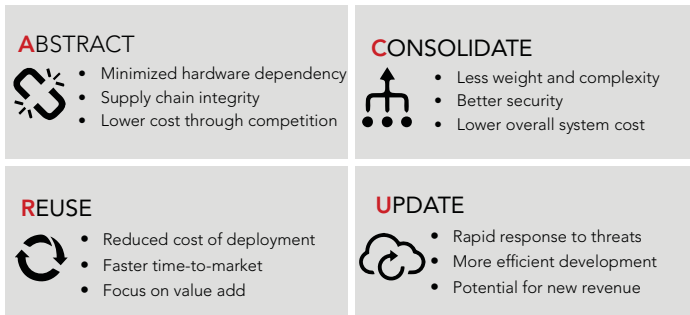


Figure 1. The ACRU model outlines four capabilities needed by OEMs to meet consumers' in-vehicle software expectations

The A of ACRU stands for abstraction, which calls for breaking the dependencies between hardware and software. Hardware is what typically drives up the cost of vehicle production, and it's to the OEM's benefit to get the maximum value out of it. This can be achieved by minimizing the dependencies to hardware. Abstraction can also maintain supply chain integrity by enabling OEMs to create an environment in which competitive parties bid for the same piece of hardware business, whether it be at the ECU level or the algorithm level, keeping total costs down.

Wind River® delivers these advantages through multilayered decoupling, which starts with silicon agnosticism—enabling integrated circuits from multiple suppliers to support the same software architecture. We also create a core framework that serves as a backbone for building multiple variants, making it possible to bring in new software algorithms and applications, whether they're from software vendors, from an OEM, or from a Tier 1 supplier. Our approach also abstracts the development process so that OEMs can easily recreate the entire software stack, simulate their hardware, simulate their use cases and inputs and outputs, and dramatically reduce the amount of development time and cost.

The C in ACRU refers to consolidation. Simply put, if OEMs can find a way to consolidate individual compute modules in a safe

and secure manner, then they can reduce the number of modules, which reduces weight, which reduces cost and complexity in delivering new services.

Today there's a tremendous amount of horsepower in compute platforms, but much of it goes unused. Wind River sees an opportunity to bring more compute power into multi-core hardware environments and run both noncritical and critical systems side by side, taking advantage of technologies such as hypervisor or virtualization, which provide time and space partitioning.

And with the proper amount of determinism and time-and-space partitioning, it is possible to update hardware independently from the middleware and the application layer, so OEMs can drive more speed in the development cycle while reducing the cost and complexity of the hardware inside the vehicle system. This same concept can also be applied to take ECU consolidation to a whole new level. OEMs can consolidate more workloads on fewer pieces of silicon, while still having the security of knowing that a breach in one compute domain does not have a direct impact on the adjacent domains. This translates to direct savings in cost and accelerates time to market.

The R in ACRU is reuse. In the same way the auto industry has learned to reuse the physical platforms or chassis from one vehicle to the next, OEMs can benefit greatly by reusing software and hardware architectures. For example, a software system framework for one model of car could be adapted and scaled across multiple brands and multiple price classes. This way OEMs get to market faster and can focus their investments on where they can add the most value. For example, they can reuse the foundation of their software, that framework in the middle, and then customize the applications that live on top, such as ADAC or IVI systems.

The U of ACRU is update, and no matter what software architecture OEMs are using, it has to be updatable. If OEMs can continuously maintain and update in order to further enhance the system, then they can continue to create new value and new opportunities to monetize.

Auto OEMs have historically been slow to adopt OTA because there's such a mix of critical and noncritical functionality; there are multiple security, safety, and compliance issues; and there is currently no framework that accommodates OTA updates for multiple interrelated services. In fact, many automakers today are hesitant to allow OTA updates for anything beyond adding a radio app.

To help OEMs overcome these challenges, Wind River takes a broader view that includes OTA. We look at software lifecycle management in its entirety, beginning at the ideation phase and continuing all the way to end of life. We make it possible to start modeling new ideas in a cloud environment rather than waiting for a physical environment, to explore new concepts in prototyped vehicles, to make changes quickly and easily, and to update the software in real time after the vehicle is deployed.

The software lifecycle management approach not only saves time on road testing and development but it also keeps the development cycle moving ahead at full speed. And that enables OEMs to reduce the overall cost structure. In addition, the lifecycle model makes the development cycle continuously iterative. Imagine being able, at the end of the vehicle's life, to access all of the data from that vehicle and use it to inform the next series of ideations. This enables automakers to truly differentiate themselves and add new value propositions going forward.

SOLVING CORE INDUSTRY CHALLENGES: EXAMPLES

In summary, software lifecycle management offers OEMs huge benefits in multiple categories. It provides security. It grants OEMs a higher level of efficiency. It gives OEMs transparency into the vehicle and its performance. And it enables OEMs to create new streams of revenue and new experiences for the consumer.

Wind River can help OEMs achieve these benefits through Wind River Helix™ Edge Sync. Edge Sync is a robust OTA update and software lifecycle management solution that enables OEMs to maintain the integrity of embedded systems and distribute performance and feature enhancements. Edge Sync also enables OEMs to collect critical data across the entire lifecycle of the vehicle, both pre- and postproduction.

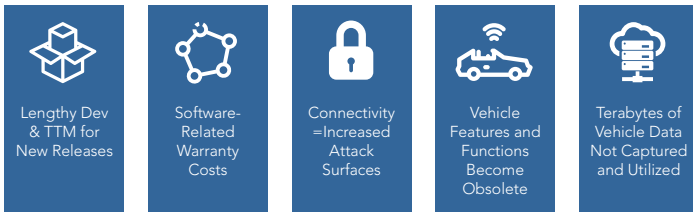


Figure 2. Industry challenges

These examples illustrate how Edge Sync solves core industry challenges:

- Cuts development and time-to-market for new software-driven services by leveraging OTA from ideation to end-of-life
- Reduces software-related warranty costs by eliminating the need for recalls by enabling updates to be done remotely
- Combats the proliferation of attack surfaces and associated cybersecurity threats by detecting and resolving security vulnerabilities
- Avoids the obsolescence of vehicle features and functions by continuously delivering new services and updating existing services
- Eliminates the “data waste” problem by aggregating and analyzing all vehicle data and turning it into actionable insights

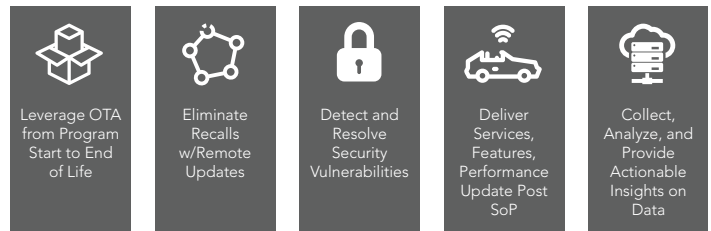


Figure 3. Software lifecycle management capabilities

CONCLUSION

It's time to take a broader perspective on the management of software in the age of the connected car. We encourage you to learn more about the comprehensive Wind River approach to software lifecycle management, from ideation to deployment, throughout the vehicle's life—and about the full spectrum of Wind River Helix Chassis solutions. We provide software products, frameworks, tools, and services to facilitate a successful transition to the architectures of the future. Let us show you how to create new economic advantages through the ACRU model.

